Package ‘itsmr’

February 20, 2015

Type Package
Title Time series analysis package for students
Version 1.5
Date 2011-11-12
Author George Weigt
Maintainer George Weigt <g808391@gmail.com>
Description This package provides a subset of the functionality found in the Windows-based program ITSM. The intended audience is students using the textbook “Introduction to Time Series and Forecasting” by Peter J. Brockwell and Richard A. Davis.
License Unlimited
LazyLoad yes
Repository CRAN
Date/Publication 2011-11-13 09:12:27
NeedsCompilation no

R topics documented:

itsmr-package ............................................................... 2
aacvf ................................................................. 3
acvf ................................................................. 5
airpass ................................................................. 6
ar.inf ................................................................. 6
arar ................................................................. 7
arma ................................................................. 8
autofit ............................................................ 9
burg ................................................................. 10
check ............................................................... 11
deaths ............................................................... 11
dowj ............................................................... 12
forecast ........................................................... 12
hannan ........................................................... 13
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>hr</td>
<td>14</td>
</tr>
<tr>
<td>ia</td>
<td>15</td>
</tr>
<tr>
<td>lake</td>
<td>16</td>
</tr>
<tr>
<td>ma.inf</td>
<td>16</td>
</tr>
<tr>
<td>periodogram</td>
<td>17</td>
</tr>
<tr>
<td>plota</td>
<td>18</td>
</tr>
<tr>
<td>plotc</td>
<td>18</td>
</tr>
<tr>
<td>plots</td>
<td>18</td>
</tr>
<tr>
<td>Resid</td>
<td>19</td>
</tr>
<tr>
<td>season</td>
<td>20</td>
</tr>
<tr>
<td>selftest</td>
<td>21</td>
</tr>
<tr>
<td>sim</td>
<td>22</td>
</tr>
<tr>
<td>smooth.exp</td>
<td>22</td>
</tr>
<tr>
<td>smooth.fft</td>
<td>23</td>
</tr>
<tr>
<td>smooth.ma</td>
<td>24</td>
</tr>
<tr>
<td>smooth.rank</td>
<td>24</td>
</tr>
<tr>
<td>specify</td>
<td>25</td>
</tr>
<tr>
<td>strikes</td>
<td>26</td>
</tr>
<tr>
<td>Sunspots</td>
<td>26</td>
</tr>
<tr>
<td>test</td>
<td>26</td>
</tr>
<tr>
<td>trend</td>
<td>27</td>
</tr>
<tr>
<td>wine</td>
<td>28</td>
</tr>
<tr>
<td>yw</td>
<td>28</td>
</tr>
</tbody>
</table>

**Index**

**itsmr-package  Time series analysis package for students**

**Description**

This package provides a subset of the functionality found in the Windows-based program ITSM. The intended audience is students using the textbook *Introduction to Time Series and Forecasting* by Peter J. Brockwell and Richard A. Davis.

**Details**

<table>
<thead>
<tr>
<th>Package</th>
<th>itsmr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>1.5</td>
</tr>
<tr>
<td>Date</td>
<td>2011-11-12</td>
</tr>
<tr>
<td>License</td>
<td>Unlimited</td>
</tr>
<tr>
<td>LazyLoad</td>
<td>yes</td>
</tr>
</tbody>
</table>
aacvf

Author(s)

George Weigt
Maintainer: George Weigt <g808391@gmail.com>

References


Examples

```r
plotc(wine)

## Define a suitable transformation of the data
xv = c("log","season",12,"trend",1)

## Obtain residuals and check for stationarity
e = resid(wine,xv)
test(e)

## Define a suitable ARMA model
a = arma(e,p=1,q=1)

## Obtain residuals and check for white noise
ee = resid(wine,xv,a)
test(ee)

## Forecast future values
forecast(wine,xv,a)
```

---

**aacvf**

*Autocovariance of ARMA model*

Description

Autocovariance of ARMA model

Usage

```r
aacvf(a, h)
```

Arguments

- **a**: ARMA model
- **h**: Maximum lag
Details

The ARMA model is a list with the following components.
acvf

\[ \text{phi} \] Vector of AR coefficients (index number equals coefficient subscript)
\[ \text{theta} \] Vector of MA coefficients (index number equals coefficient subscript)
\[ \text{sigma2} \] White noise variance

Value

Returns a vector of length \( h+1 \) to accommodate lag 0 at index 1.

See Also

arma

Examples

\[ a = \text{arma(Sunspots,2,0)} \]
\[ \text{acvf}(a,40) \]

\[
\begin{array}{c|c}
\text{acvf} & \text{Autocovariance of data} \\
\end{array}
\]

Description

Autocovariance of data

Usage

\[ \text{acvf}(x, h = 40) \]

Arguments

\[ x \] Data vector
\[ h \] Maximum lag

Value

Returns a vector of length \( h+1 \) to accommodate lag 0 at index 1.

See Also

plota

Examples

\[ \text{acvf(Sunspots)} \]
airpass

*Number of international airline passengers, 1949 to 1960*

**Description**

Number of international airline passengers, 1949 to 1960

**Examples**

```
plotc(airpass)
```

**ar.inf**

*Compute AR infinity coefficients*

**Description**

Compute AR infinity coefficients

**Usage**

```
ar.inf(a, n = 50)
```

**Arguments**

- `a`: ARMA model
- `n`: Order

**Details**

The ARMA model is a list with the following components.

- `phi`: Vector of AR coefficients (index number equals coefficient subscript)
- `theta`: Vector of MA coefficients (index number equals coefficient subscript)
- `sigma2`: White noise variance

**Value**

Returns a vector of length \( n+1 \) to accommodate coefficient 0 at index 1.

**See Also**

`ma.inf`
**Examples**

\[
a = yw(Sunspots,2) \\
ar.\text{inf}(a)
\]

**Description**

Forecast using ARAR algorithm

**Usage**

\[
arar(y, h = 10, opt = 2)
\]

**Arguments**

- `y` Data vector
- `h` Steps ahead
- `opt` Display option (0 silent, 1 tabulate, 2 plot and tabulate)

**Value**

Returns the following list invisibly.

- `pred` Predicted values
- `se` Standard errors
- `l` Lower bounds (95% confidence interval)
- `u` Upper bounds

**See Also**

- `forecast`

**Examples**

\[
arar(airpass)
\]
 arma

Estimate ARMA model coefficients using maximum likelihood

Description

Estimate ARMA model coefficients using maximum likelihood

Usage

arma(x, p = 0, q = 0)

Arguments

  x     Data vector
  p     AR order
  q     MA order

Details

Calls the standard R function arima to estimate AR and MA coefficients. The innovations algorithm is used to estimate white noise variance.

Value

Returns an ARMA model consisting of a list with the following components.

  phi    Vector of AR coefficients (index number equals coefficient subscript)
  theta  Vector of MA coefficients (index number equals coefficient subscript)
  sigma2 White noise variance
  aicc   Akaike information criterion corrected
  se.phi Standard errors for the AR coefficients
  se.theta Standard errors for the MA coefficients

See Also

  autofit burg hannan ia yw

Examples

  xv = c("diff",1)
  e = Resid(dowj,xv)
  a = arma(e,1,0)
  print(a)
autofit  

Find the best model from a range of possible ARMA models

Description

Find the best model from a range of possible ARMA models

Usage

autofit(x, p = 0:5, q = 0:5)

Arguments

- **x**: Data vector (typically residuals from Resid)
- **p**: Range of AR orders
- **q**: Range of MA orders

Details

Tries all combinations of p and q and returns the model with the lowest AICC. The arguments p and q should be small ranges as this function can be slow otherwise. The innovations algorithm is used to estimate white noise variance.

Value

Returns an ARMA model consisting of a list with the following components.

- **phi**: Vector of AR coefficients (index number equals coefficient subscript)
- **theta**: Vector of MA coefficients (index number equals coefficient subscript)
- **sigma2**: White noise variance
- **aicc**: Akaike information criterion corrected
- **se.phi**: Standard errors for the AR coefficients
- **se.theta**: Standard errors for the MA coefficients

See Also

arma

Examples

```r
xv = c("diff",1)
e = Resid(dowj,xv)
a = autofit(e)
print(a)
```
burg

Estimate AR coefficients using the Burg method

Description

Estimate AR coefficients using the Burg method

Usage

burg(x, p)

Arguments

  x  Data vector (typically residuals from resid)
  p  AR order

Details

The innovations algorithm is used to estimate white noise variance.

Value

Returns an ARMA model consisting of a list with the following components.

phi  Vector of AR coefficients (index number equals coefficient subscript)
theta 0
sigma2  White noise variance
aicc  Akaike information criterion corrected
se.phi  Standard errors for the AR coefficients
se.theta 0

See Also

arma hanan ia yw

Examples

  xv = c("diff",1)
  e = Resid(dowj,xv)
  a = burg(e,1)
  print(a)
check  
*Check for causality and invertibility*

**Description**
Check for causality and invertibility

**Usage**
check(a)

**Arguments**
a  ARMA model

**Details**
The ARMA model is a list with the following components.

- **phi**  Vector of AR coefficients (index number equals coefficient subscript)
- **theta**  Vector of MA coefficients (index number equals coefficient subscript)
- **sigma2**  White noise variance

**Value**
None

**Examples**
```r
a = specify(ar=c(0,0,.99))
check(a)
```

---

**deaths**  
*USA accidental deaths, 1973 to 1978*

**Description**
USA accidental deaths, 1973 to 1978

**Examples**
```r
plotc(deaths)
```
**Description**

Dow Jones utilities index, August 28 to December 18, 1972

**Examples**

```r
dowj
```  

**forecast**

*Forecast future values*

**Description**

Forecast future values

**Usage**

```r
forecast(x, xv, a, h = 10, opt = 2)
```  

**Arguments**

- `x` Data vector
- `xv` Transform vector
- `a` ARMA model
- `h` Steps ahead
- `opt` Display option (0 silent, 1 tabulate, 2 plot and tabulate)

**Details**

The transform vector can be NULL for none. Otherwise `xv` is a vector that specifies a sequence of transform functions.

Example:

```r
xv = c("log","season",12,"trend",1)
```

The above transform vector takes the log of the data, then subtracts a seasonal component of period 12, then subtracts a linear trend component.

There are five functions from which to choose.

- `diff` Difference the data. Has a single argument, the lag.
- `hr` Subtract harmonic components. Has one or more arguments, each specifying the number of observations per harmonic.
- `log` Take the log of the data, has no arguments.
- `season` Subtract a seasonal component. Has a single argument, the number of observations per season.
- `trend` Subtract a trend component. Has a single argument, the order of the trend (1 linear, 2 quadratic, etc.)
At the end of the transform vector there is an implied subtraction of the mean operation. Hence the resulting time series always has zero mean.

All of the transformations are inverted before the forecast results are displayed.

Value

Returns the following list invisibly.

- `pred`: Predicted values
- `se`: Standard errors (not included if there is a log transform)
- `l`: Lower bounds (95% confidence interval)
- `u`: Upper bounds

See Also

`arma`, `Resid`, `test`

Examples

```r
xv = c("log", "season", 12, "trend", 1)
e = Resid(wine, xv)
a = arma(e, 1, 1)
forecast(wine, xv, a)
```

---

### hannan

**Estimate ARMA coefficients using the Hannan-Rissanen algorithm**

**Description**

Estimate ARMA coefficients using the Hannan-Rissanen algorithm

**Usage**

```r
hannan(x, p, q)
```

**Arguments**

- `x`: Data vector (typically residuals from Resid)
- `p`: AR order
- `q`: MA order (q > 0)

**Details**

The innovations algorithm is used to estimate white noise variance.
Value

Returns an ARMA model consisting of a list with the following components.

- **phi**: Vector of AR coefficients (index number equals coefficient subscript)
- **theta**: Vector of MA coefficients (index number equals coefficient subscript)
- **sigma2**: White noise variance
- **aicc**: Akaike information criterion corrected
- **se.phi**: Standard errors for the AR coefficients
- **se.theta**: Standard errors for the MA coefficients

See Also

*arma burg ia yw*

Examples

```r
xv = c("diff", 12)
e = resid(deaths, xv)
a = hannan(e, 1, 1)
print(a)
```

---

hr  

*Estimate harmonic components*

Description

Estimate harmonic components

Usage

```r
hr(x, d)
```

Arguments

- **x**: Data vector
- **d**: Vector of harmonic periods

Value

Returns a vector the same length as x. Subtract from x to obtain residuals.

Examples

```r
y = hr(deaths, c(12, 6))
plotc(deaths, y)
```
Description

Estimate MA coefficients using the innovations algorithm

Usage

ia(x, q, m = 17)

Arguments

x Data vector (typically residuals from Resid)
q MA order
m Recursion level

Details

Normally m should be set to the default value. The innovations algorithm is used to estimate white noise variance.

Value

Returns an ARMA model consisting of a list with the following components.

phi 0
theta Vector of MA coefficients (index number equals coefficient subscript)
sigma2 White noise variance
aicc Akaike information criterion corrected
se.phi 0
se.theta Standard errors for the MA coefficients

See Also

arma burg hannan yw

Examples

xv = c("diff",1)
e = Resid(dowj,xv)
a = ia(e,1)
print(a)
lake  

Description

Level of Lake Huron, 1875 to 1972

Examples

plotc(lake)

ma.inf  

Description

Compute MA infinity coefficients

Usage

ma.inf(a, n = 50)

Arguments

a ARMA model
n Order

Details

The ARMA model is a list with the following components.

phi Vector of AR coefficients (index number equals coefficient subscript)
theta Vector of MA coefficients (index number equals coefficient subscript)
sigma2 White noise variance

Value

Returns a vector of length n+1 to accommodate coefficient 0 at index 1.

See Also

ar.inf
periodogram

Examples

```r
xv = c("diff",12)
e = Resid(deaths,xv)
a = arma(e,1,1)
ma.inf(a,10)
```

periodogram | Plot a periodogram

Description

Plot a periodogram

Usage

```r
periodogram(x, q = 0, opt = 2)
```

Arguments

- `x`: Data vector
- `q`: MA filter order
- `opt`: Plot option (0 silent, 1 periodogram only, 2 periodogram and filter)

Details

The filter `q` can be a vector in which case the overall filter is the composition of MA filters of the designated orders.

Value

The periodogram vector divided by 2pi is returned invisibly.

See Also

- plots

Examples

```r
periodogram(Sunspots,c(1,1,1,1))
```
**plota**

*Plot data and/or model ACF and PACF*

**Description**
Plot data and/or model ACF and PACF

**Usage**
```
plota(u, v = NULL, h = 40)
```

**Arguments**
- `u, v` Data and/or ARMA model in either order
- `h` Maximum lag

**Value**
None

**Examples**
```
plota(Sunspots)
a = yw(Sunspots, 2)
plota(Sunspots, a)
```

**plotc**

*Plot one or two time series*

**Description**
Plot one or two time series

**Usage**
```
plotc(y1, y2 = NULL)
```

**Arguments**
- `y1` Data vector (plotted in blue with knots)
- `y2` Data vector (plotted in red, no knots)

**Value**
None
**plots**

Plot spectrum of data or ARMA model

**Description**
Plot spectrum of data or ARMA model

**Usage**
`plots(u)`

**Arguments**
- `u` Data vector or an ARMA model

**Value**
None

**See Also**
`periodogram`

**Examples**
```r
a = specify(ar=c(0,0,.99))
plots(a)
```

---

**Resid**
Compute residuals

**Description**
Compute residuals

**Usage**
`Resid(x, xv = NULL, a = NULL)`
Arguments

- **x**: Data vector
- **xv**: Transform vector
- **a**: ARMA model

Details

The transform vector can be NULL for none. Otherwise `xv` is a vector that specifies a sequence of transform functions.

Example:

```r
xv = c("log", "season", 12, "trend", 1)
```

The above transform vector takes the log of the data, then subtracts a seasonal component of period 12, then subtracts a linear trend component.

There are five functions from which to choose:

- **diff**: Difference the data. Has a single argument, the lag.
- **hr**: Subtract harmonic components. Has one or more arguments, each specifying the number of observations per harmonic.
- **log**: Take the log of the data, has no arguments.
- **season**: Subtract a seasonal component. Has a single argument, the number of observations per season.
- **trend**: Subtract a trend component. Has a single argument, the order of the trend (1 linear, 2 quadratic, etc.)

At the end of the transform vector there is an implied subtraction of the mean operation. Hence the resulting time series always has zero mean.

Value

Returns a vector of residuals the same length as `x`.

See Also

- `test`

Examples

```r
xv = c("log", "season", 12, "trend", 1)
e = Resid(wine, xv)
a = arma(e, 1, 1)
ee = Resid(wine, xv, a)
```

---

**season**

Estimate seasonal component

Description

Estimate seasonal component
selftest

Usage
season(x, d)

Arguments
x Data vector
d Number of observations per season

Value
Returns a vector the same length as x. Subtract from x to obtain residuals.

See Also
trend

Examples
y = season(deaths, 12)
plotc(deaths, y)

---

selftest  Run a self test

Description
Run a self test

Usage
selftest()

Details
This function is a useful check if the code is modified.

Value
None

Examples
selftest()
**Sim**

*Generate synthetic observations*

**Description**

Generate synthetic observations

**Usage**

```r
sim(a, n = 100)
```

**Arguments**

- `a` ARMA model
- `n` Number of synthetic observations required

**Details**

The ARMA model is a list with the following components.

- `phi` Vector of AR coefficients (index number equals coefficient subscript)
- `theta` Vector of MA coefficients (index number equals coefficient subscript)
- `sigma2` White noise variance

**Value**

Returns a vector of `n` synthetic observations.

**Examples**

```r
a = specify(ar=c(0,0,.99))
x = sim(a,60)
plotc(x)
```

---

**Smooth.exp**

*Apply an exponential filter*

**Description**

Apply an exponential filter

**Usage**

```r
smooth.exp(x, alpha)
```
smooth.fft

Arguments

  x         Data vector
  alpha     Smoothness setting, 0-1

Details

  Zero is maximum smoothness.

Value

  Returns a vector of smoothed data the same length as x.

Examples

  y = smooth.exp(strikes, .4)
  plotc(strikes, y)

---

smooth.fft          Apply a low pass filter

Description

  Apply a low pass filter

Usage

  smooth.fft(x, f)

Arguments

  x         Data vector
  f         Cut-off frequency, 0-1

Details

  The cut-off frequency is specified as a fraction. For example, c=.25 passes the lowest 25\% of the spectrum.

Value

  Returns a vector the same length as x.

Examples

  y = smooth.fft(deaths, .1)
  plotc(deaths, y)
### smooth.ma

**Apply a moving average filter**

**Description**

Apply a moving average filter

**Usage**

```
smooth.ma(x, q)
```

**Arguments**

- `x`: Data vector
- `q`: Filter order

**Details**

The averaging function uses $2q+1$ values.

**Value**

Returns a vector the same length as `x`.

**Examples**

```
y = smooth.ma(strikes, 2)
plotc(strikes, y)
```

---

### smooth.rank

**Apply a spectral filter**

**Description**

Apply a spectral filter

**Usage**

```
smooth.rank(x, k)
```

**Arguments**

- `x`: Data vector
- `k`: Number of frequencies
Details
Passes the mean and the k frequencies with the highest amplitude. The remainder of the spectrum is filtered out.

Value
Returns a vector the same length as x.

Examples
\[
y = \text{smooth.rank}(\text{deaths}, 2) \\
\text{plotc}(\text{deaths}, y)
\]

---

**specify**  
*Specify an ARMA model*

Description
Specify an ARMA model

Usage
```r
specify(ar = 0L, ma = 0, sigma2 = 1)
```

Arguments
- **ar**  
  Vector of AR coefficients (index number equals coefficient subscript)
- **ma**  
  Vector of MA coefficients (index number equals coefficient subscript)
- **sigma2**  
  White noise variance

Value
Returns an ARMA model consisting of a list with the following components.

- **phi**  
  Vector of AR coefficients (index number equals coefficient subscript)
- **theta**  
  Vector of MA coefficients (index number equals coefficient subscript)
- **sigma2**  
  White noise variance

Examples
```r
specify(ar = c(0, 0, .99))
```
strikes  
USA union strikes, 1951-1980  

Description  
USA union strikes, 1951-1980  

Examples  
plotc(strikes)  

Sunspots  
Number of sunspots, 1770 to 1869  

Description  
Number of sunspots, 1770 to 1869  

Examples  
plotc(Sunspots)  

test  
Test residuals for stationarity and randomness  

Description  
Test residuals for stationarity and randomness  

Usage  
test(e)  

Arguments  
e  Data vector (typically residuals from Resid)  

Details  
Plots ACF, PACF, residuals, and QQ. Displays results for Ljung-Box, McLeod-Li, turning point, difference-sign, and rank tests. The plots can be used to check for stationarity and the other tests check for white noise.
trend

Value

None

See Also

Resid

Examples

```r
xv = c("log", "season", 12, "trend", 1)
e = Resid(wine, xv)
test(e) ## Is e stationary?
a = arma(e, 1, 1)
 ee = Resid(wine, xv, a)
test( ee) ## Is ee white noise?
```

trend | Estimate trend component

Description

Estimate trend component

Usage

```r
trend(x, p)
```

Arguments

<table>
<thead>
<tr>
<th>x</th>
<th>Data vector</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>Polynomial order (1 linear, 2 quadratic, etc.)</td>
</tr>
</tbody>
</table>

Value

Returns a vector the same length as `x`. Subtract from `x` to obtain residuals. The returned vector is the least squares fit of a polynomial to the data.

See Also

season

Examples

```r
y = trend(uspop, 2)
plotc(uspop, y)
```
wine

Australian red wine sales, January 1980 to October 1991

Description

Australian red wine sales, January 1980 to October 1991

Examples

plotc(wine)

yw

Estimate AR coefficients using the Yule-Walker method

Description

Estimate AR coefficients using the Yule-Walker method

Usage

yw(x, p)

Arguments

x  Data vector (typically residuals from Resid)

p  AR order

Details

The innovations algorithm is used to estimate white noise variance.

Value

Returns an ARMA model consisting of a list with the following components.

phi  Vector of AR coefficients (index number equals coefficient subscript)
theta  0

sigma2  White noise variance

aicc  Akaike information criterion corrected

se.phi  Standard errors for the AR coefficients

se.theta  0

See Also

arma burg hannan ia
Examples

```r
xv = c("diff",1)
e = Resid(dowj,xv)
a = yw(e,1)
```
Index

*Topic datasets
  airpass, 6
  deaths, 11
  dowj, 12
  lake, 16
  strikes, 26
  Sunspots, 26
  wine, 28

*Topic package
  itsmr-package, 2

aacvf, 3
acvf, 5
airpass, 6
ar.inf, 6, 16
arar, 7
arma, 5, 8, 9, 10, 13–15, 28
autofit, 8, 9

burg, 8, 10, 14, 15, 28

check, 11

deaths, 11

dowj, 12

forecast, 7, 12

hannan, 8, 10, 13, 15, 28
hr, 14

ia, 8, 10, 14, 15, 28

itsmr (itsmr-package), 2

itsmr-package, 2

lake, 16

ma.inf, 6, 16

periodogram, 17, 19

plota, 5, 18

plotc, 18
plots, 17, 19

Resid, 13, 19, 27

season, 20, 27

selftest, 21

sim, 22
smooth.exp, 22
smooth.fft, 23
smooth.ma, 24
smooth.rank, 24
specify, 25
strikes, 26

Sunspots, 26
test, 13, 20, 26
trend, 21, 27

wine, 28

yw, 8, 10, 14, 15, 28